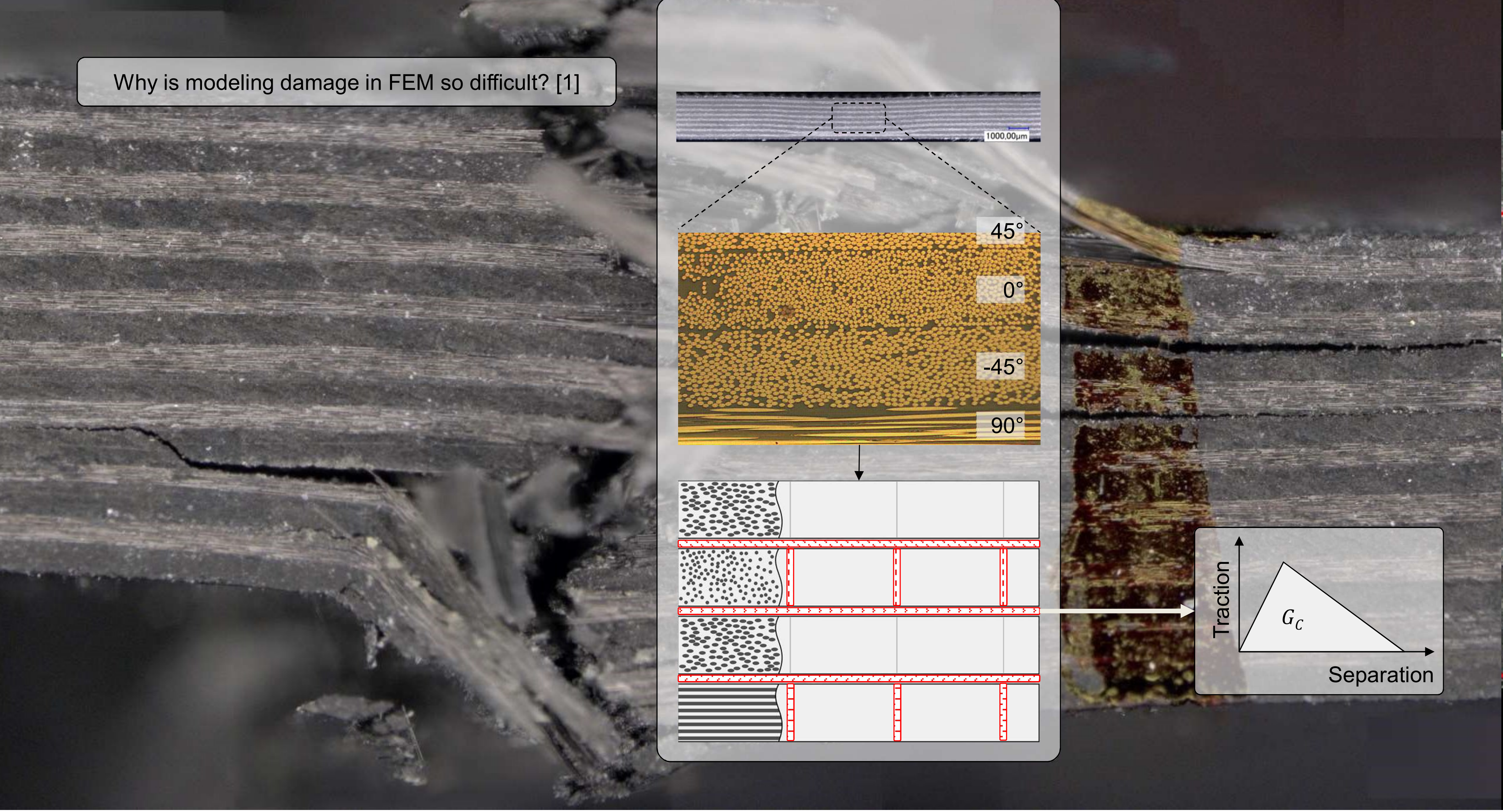
PERIDYNAMIC SIMULATION PLATTFORM TO DETERMINE VIRTUAL ALLOWABLES OF MANUFACTURING DEVIATIONS

Christian Willberg, Jan-Timo Hesse, Daniel Krause

DLR

Why is modeling damage in FEM so difficult? [1]



Damage – part of the problem or the solution?

Continuum mechanics / *Peridynamics*

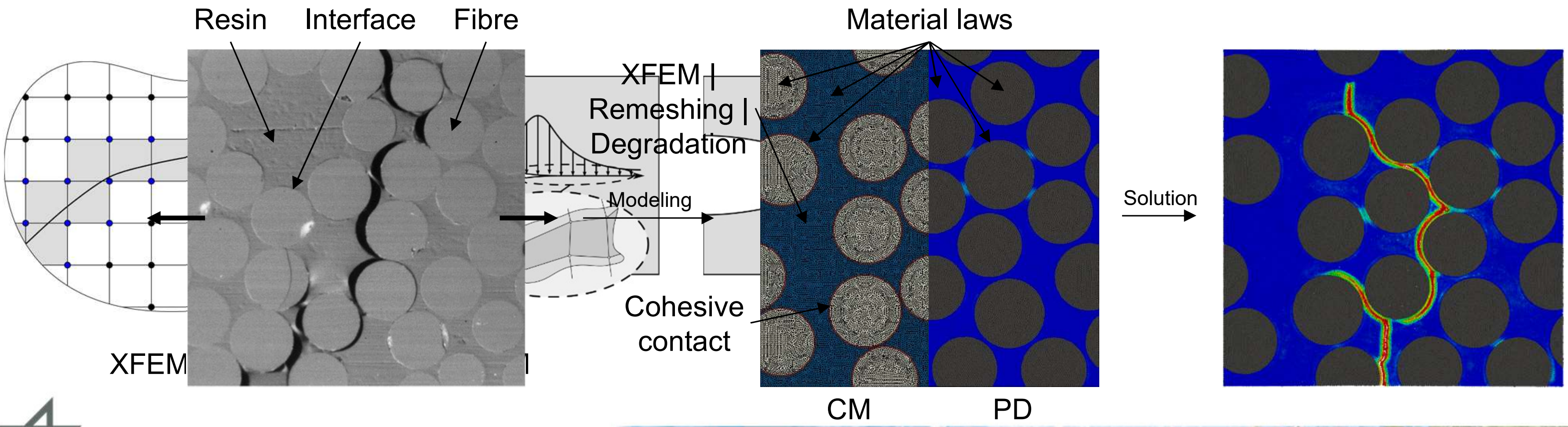
Continuum mechanics (CM) & FEM

- Assumptions:
  - Continuous medium ↯
  - $\boldsymbol{u}$ 2x continuously differentiable ↯
  - Conservation equations satisfied
  - …
- Momentum conservation: $\nabla\boldsymbol{\sigma} + \boldsymbol{b} = \rho\ddot{\boldsymbol{u}}$ ↯

Peridynamics (PD)

- Assumption:
  - Conservation equations satisfied
- Momentum conservation:

$$\int_{\delta} \left[\underline{\boldsymbol{T}}(\boldsymbol{x},t)\langle\boldsymbol{q}-\boldsymbol{x}\rangle - \underline{\boldsymbol{T}}(\boldsymbol{q},t)\langle\boldsymbol{x}-\boldsymbol{q}\rangle dV_{\boldsymbol{q}}\right] + \boldsymbol{b} = \rho\ddot{\boldsymbol{u}}$$



Modeling

$$\underline{\mathbf{T}}\langle\boldsymbol{\xi}\rangle = \underline{\omega}\langle\boldsymbol{\xi}\rangle\mathbf{P}\mathbf{K}^{-1}\boldsymbol{\xi}$$

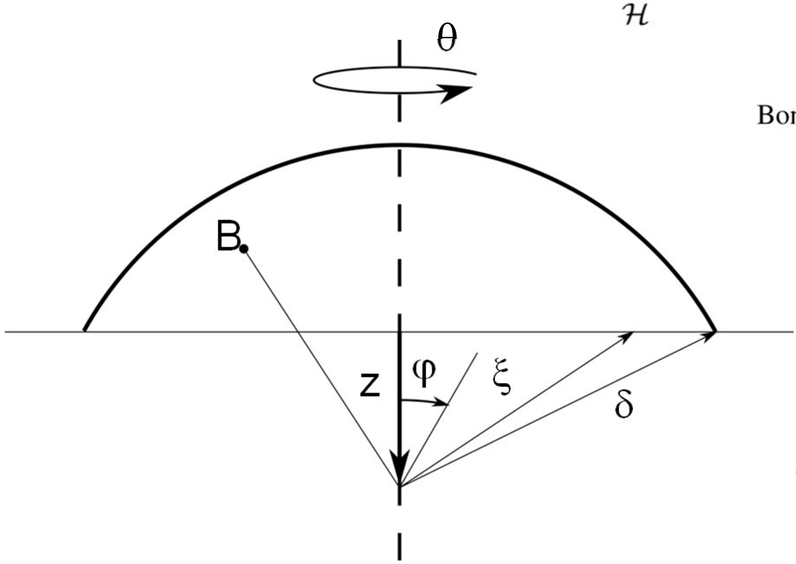
$$\mathbf{P} = \det\mathbf{F}\boldsymbol{\sigma}\mathbf{F}^{-1}$$

$$\mathbf{F} = \left[\int_{\mathcal{H}} \underline{\omega}\langle\boldsymbol{\xi}\rangle\underline{\mathbf{Y}}\langle\boldsymbol{\xi}\rangle \otimes \underline{\mathbf{X}}\langle\boldsymbol{\xi}\rangle dV_x\boldsymbol{\xi}\right] \cdot \mathbf{K}^{-1}$$

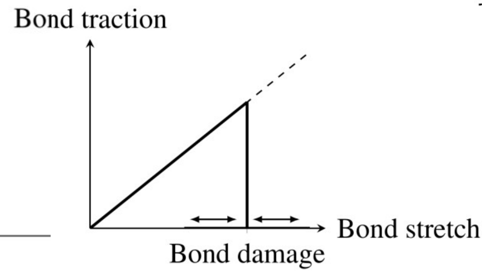
$$\boldsymbol{\sigma} = f\left(\frac{\partial\boldsymbol{\sigma}}{\partial\boldsymbol{\varepsilon}}, \boldsymbol{\varepsilon}\right)$$

$$\mathbf{K} = \int_{\mathcal{H}} \underline{\omega}\langle\boldsymbol{\xi}\rangle\underline{\mathbf{X}}\langle\boldsymbol{\xi}\rangle \otimes \underline{\mathbf{X}}\langle\boldsymbol{\xi}\rangle dV_x\boldsymbol{\xi}$$

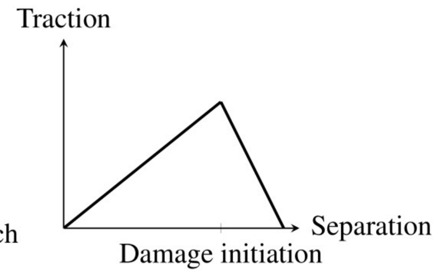
User Material



Local



Global

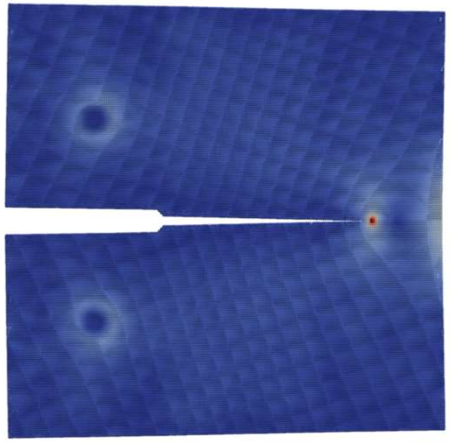


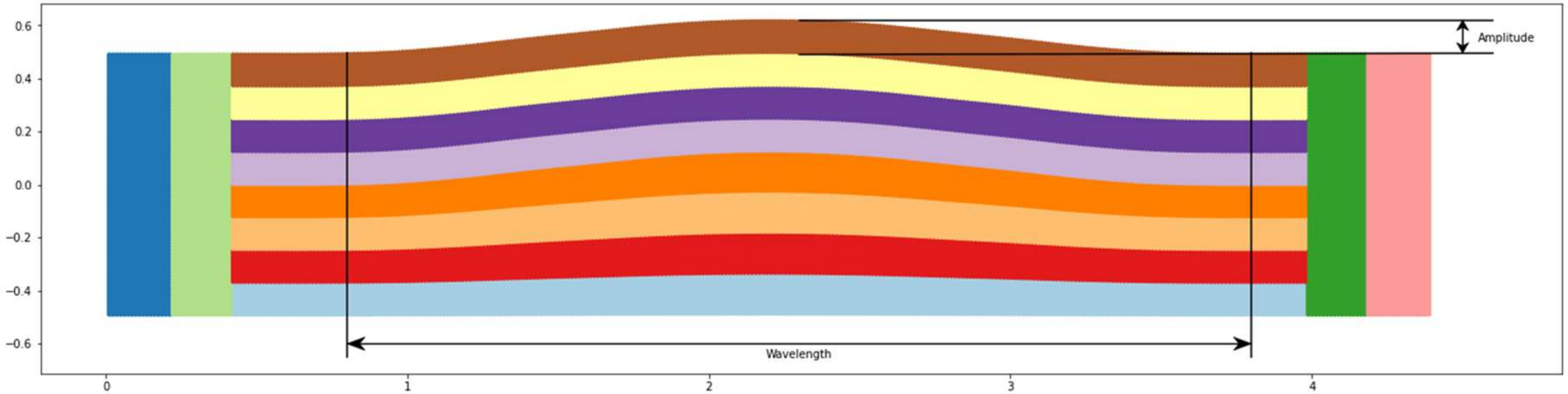
$$G = 2\int_0^{\delta}\int_{H_r} w_b dV_\xi$$

$$w_b = \frac{4G}{\pi\delta^4}$$

$$w_b > w_{critical}$$

Virtual Testing

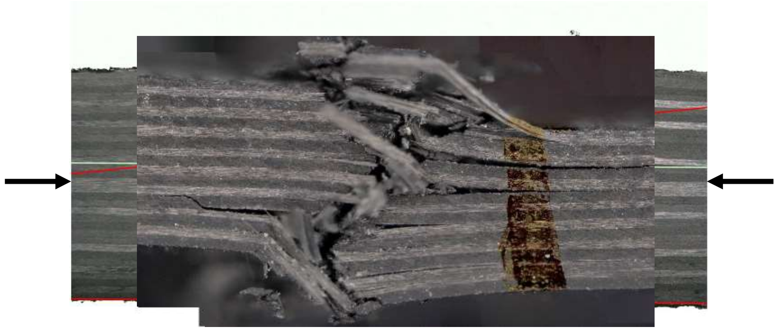


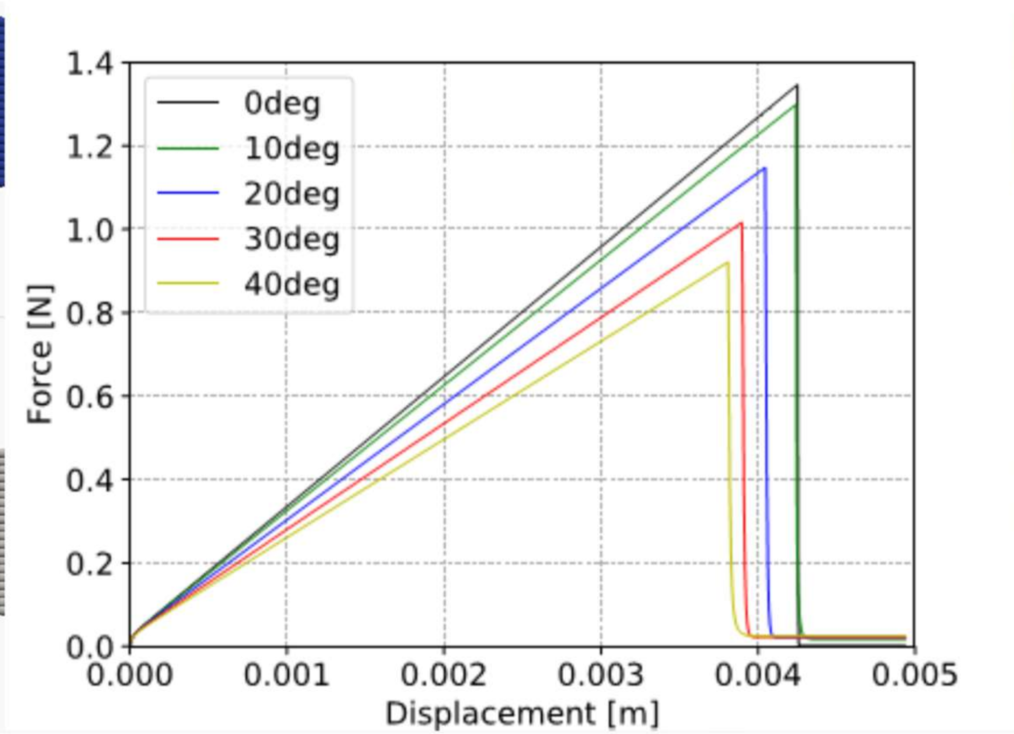


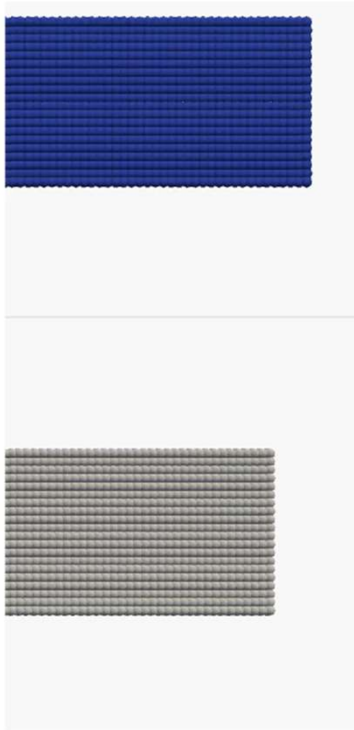
Peridynamics

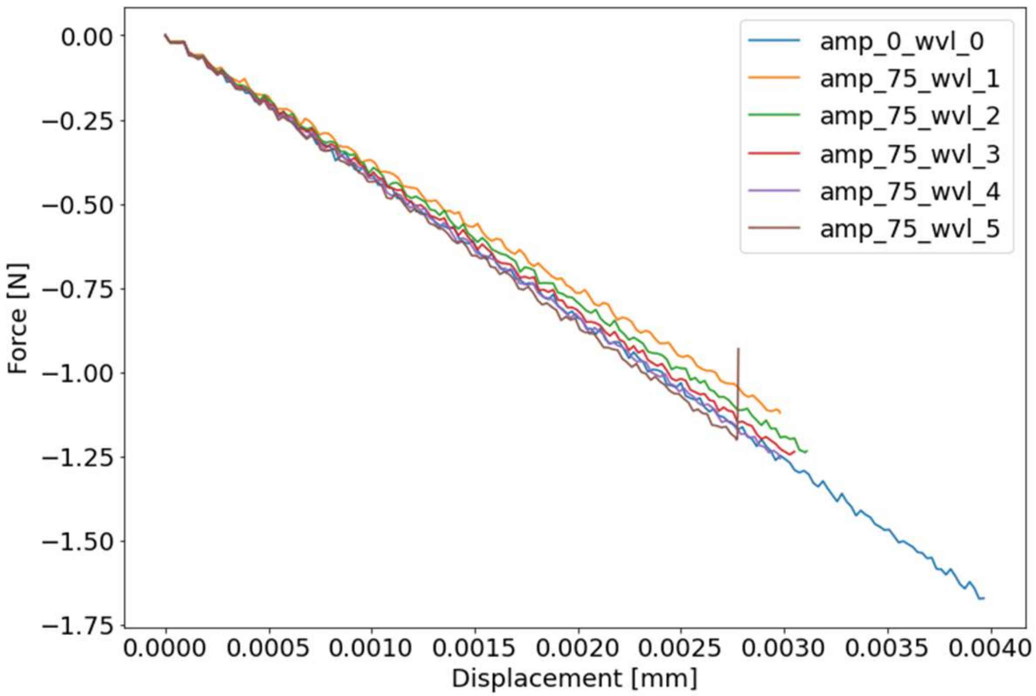
Numerical prediction of phenomena

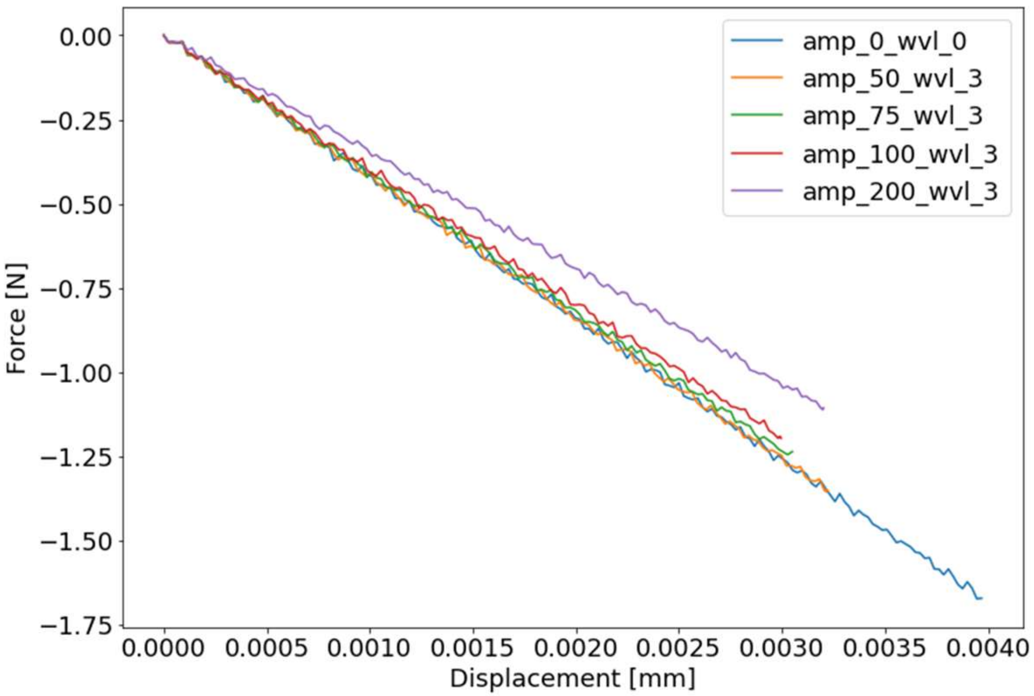
- Example: Waviness
- Modeled by local variation of material orientation

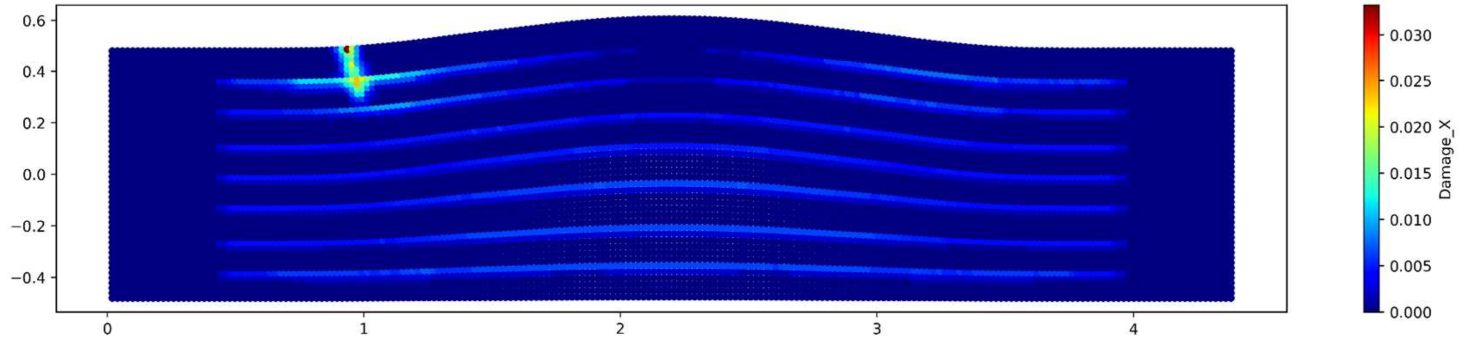







amp_0_wvl_0
amp_75_wvl_1
amp_75_wvl_2
amp_75_wvl_3
amp_75_wvl_4
amp_75_wvl_5
Force [N]
Displacement [mm]


amp_0_wvl_0
amp_50_wvl_3
amp_75_wvl_3
amp_100_wvl_3
amp_200_wvl_3
Force [N]
Displacement [mm]


Damage_X

Conclusion

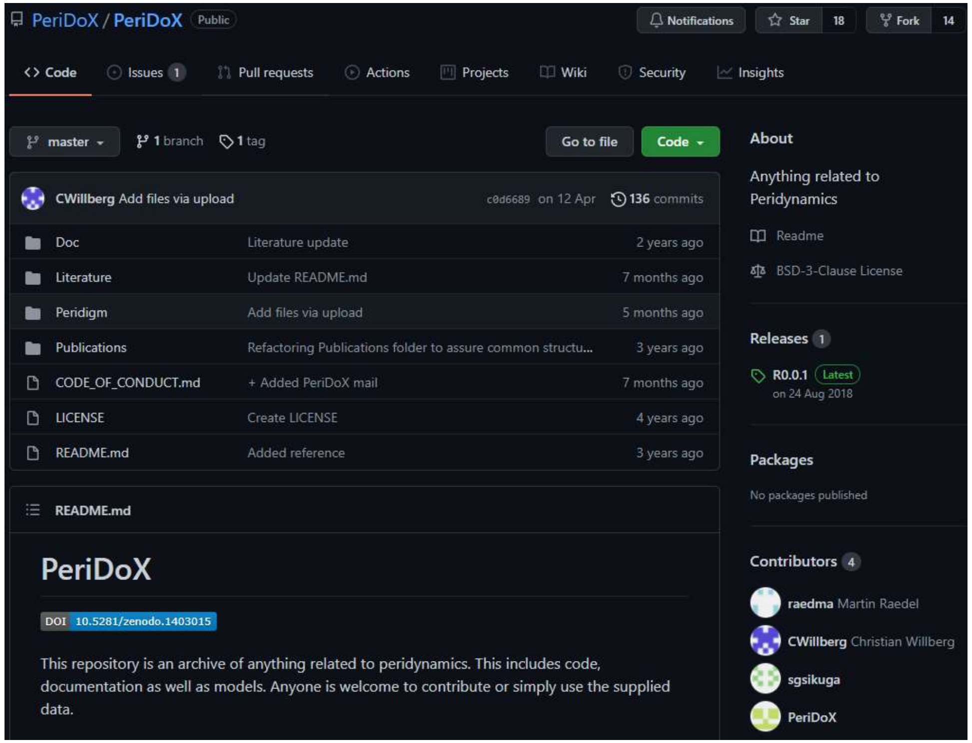
- Ondulations have significant influence in load carrying capacity
- Virtual testing helps to catch a wide varity of scenarios
- Peridynamics is a good method to decribe the progressive failure

- Next steps
    - Analysis of a wide range of parameters and sensitivity analysis
    - Validation of the process

Bibliography

Raffael Bogenfeld, Janko Kreikemeier, and Tobias Wille. “Review and benchmark study on the analysis of low-velocity impact on composite laminates”. In: Engineering Failure Analysis 86 (2018), pp. 72–99. ISSN : 1350-6307. DOI : 10.1016/j.engfailanal.2017.12.019

[2] Daniel Krause. “Micromechanics of the fatigue behaviour of polymers”. DLR Report 2016-26. PhD Thesis. Technical University Braunschweig, 2016.



https://github.com/PeriDoX/PeriDoX

Acknowledgement

The work was done in German Research Foundation funded project: “Gekoppelte Peridynamik-Finite-Elemente-Simulationen zur Schädigungsanalyse von Faserverbundstrukturen”

Grant number: WI **4835/5-1**

and the

M-ERA.NET funded project “Exploring Multi-Method Analysis of composite structures and joints under consideration of uncertainties engineering and processing (EMMA)“

This measure is co-financed with tax funds on the basis of the budget passed by the Saxon state parlament. Grant number: **3028223**.

Thank you!

Dr.-Ing. Christian Willberg

DLR
Institute of Composite Structures
and Adaptive Systems

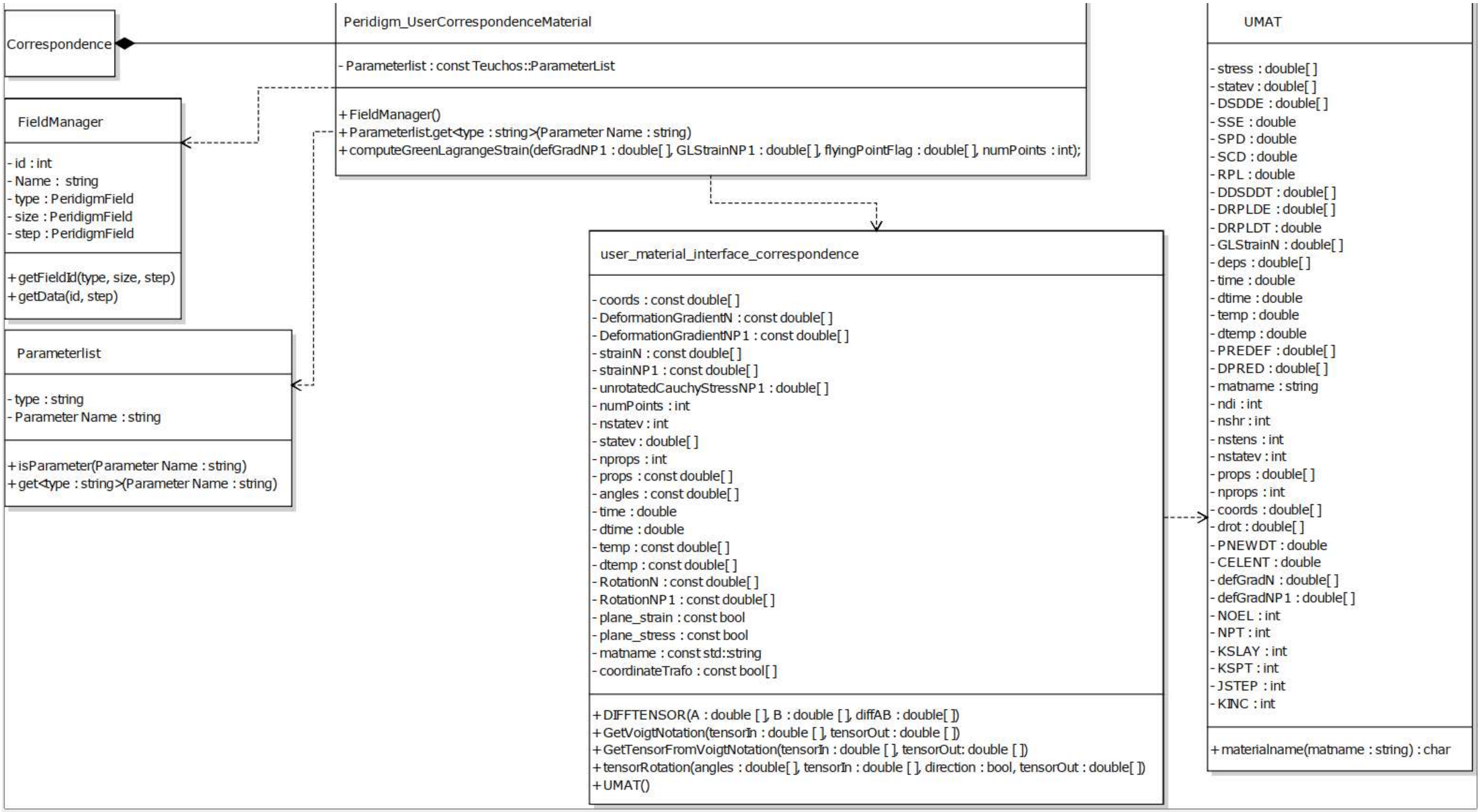
Lilienthalplatz 7
38108 Braunschweig
Germany

Phone: +49 531 295 - 2232
Email: christian.willberg@dlr.de

For further information

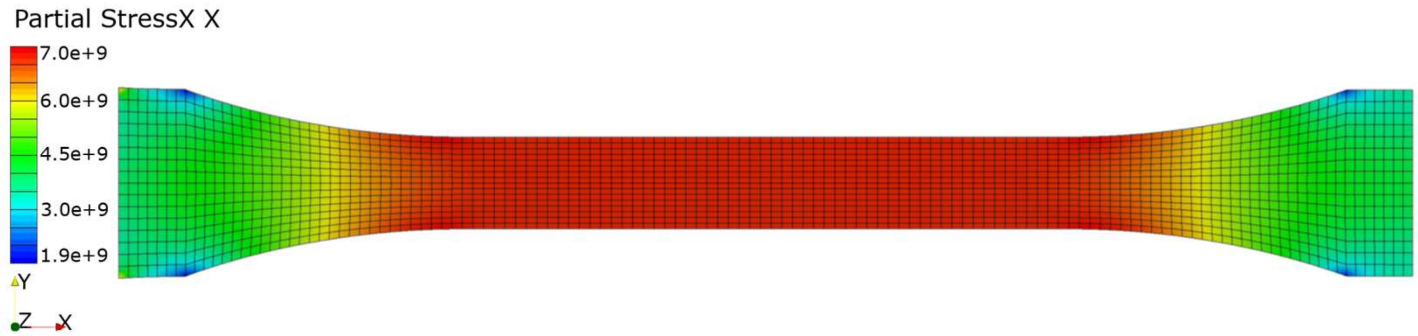
Rädel, M. & Willberg, C. PeriDoX Repository
https://github.com/PeriDoX/PeriDoX
Doi: 10.5281/zenodo.1403015

DLR

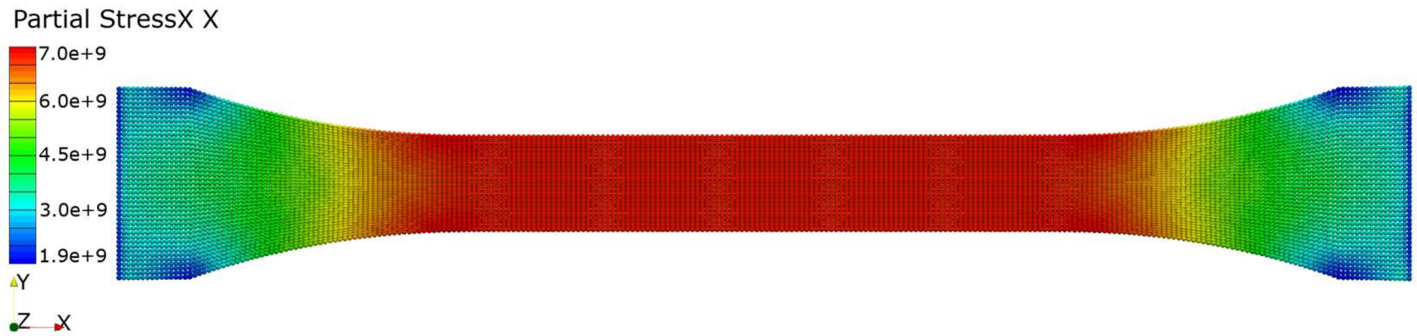

Correspondence
Peridigm_UserCorrespondenceMaterial
- Parameterlist : const Teuchos::ParameterList
+ FieldManager()
+ Parameterlist.get<type : string>(Parameter Name : string)
+ computeGreenLagrangeStrain(defGradNP1 : double[ ], GLStrainNP1 : double[ ], flyingPointFlag : double[ ], numPoints : int);
UMAT
- stress : double[ ]
- statev : double[ ]
- DSDDE : double[ ]
- SSE : double
- SPD : double
- SCD : double
- RPL : double
- DDSDDT : double[ ]
- DRPLDE : double[ ]
- DRPLDT : double
- GLStrainN : double[ ]
- deps : double[ ]
- time : double
- dtime : double
- temp : double
- dtemp : double
- PREDEF : double[ ]
- DPRED : double[ ]
- matname : string
- ndi : int
- nshr : int
- nstens : int
- nstatev : int
- props : double[ ]
- nprops : int
- coords : double[ ]
- drot : double[ ]
- PNEWDT : double
- CELENT : double
- defGradN : double[ ]
- defGradNP1 : double[ ]
- NOEL : int
- NPT : int
- KSLAY : int
- KSPT : int
- JSTEP : int
- KINC : int
+ materialname(matname : string) : char
FieldManager
- id : int
- Name : string
- type : PeridigmField
- size : PeridigmField
- step : PeridigmField
+ getFieldId(type, size, step)
+ getData(id, step)
Parameterlist
- type : string
- Parameter Name : string
+ isParameter(Parameter Name : string)
+ get<type : string>(Parameter Name : string)
user_material_interface_correspondence
- coords : const double[ ]
- DeformationGradientN : const double[ ]
- DeformationGradientNP1 : const double[ ]
- strainN : const double[ ]
- strainNP1 : const double[ ]
- unrotatedCauchyStressNP1 : double[ ]
- numPoints : int
- nstatev : int
- statev : double[ ]
- nprops : int
- props : const double[ ]
- angles : const double[ ]
- time : double
- dtime : double
- temp : const double[ ]
- dtemp : const double[ ]
- RotationN : const double[ ]
- RotationNP1 : const double[ ]
- plane_strain : const bool
- plane_stress : const bool
- matname : const std::string
- coordinateTrafo : const bool[ ]
+ DIFFTENSOR(A : double [ ], B : double [ ], diffAB : double[ ])
+ GetVoigtNotation(tensorIn : double [ ], tensorOut : double [ ])
+ GetTensorFromVoigtNotation(tensorIn : double [ ], tensorOut: double [ ])
+ tensorRotation(angles : double[ ], tensorIn : double [ ], direction : bool, tensorOut : double[ ])
+ UMAT()

Verification

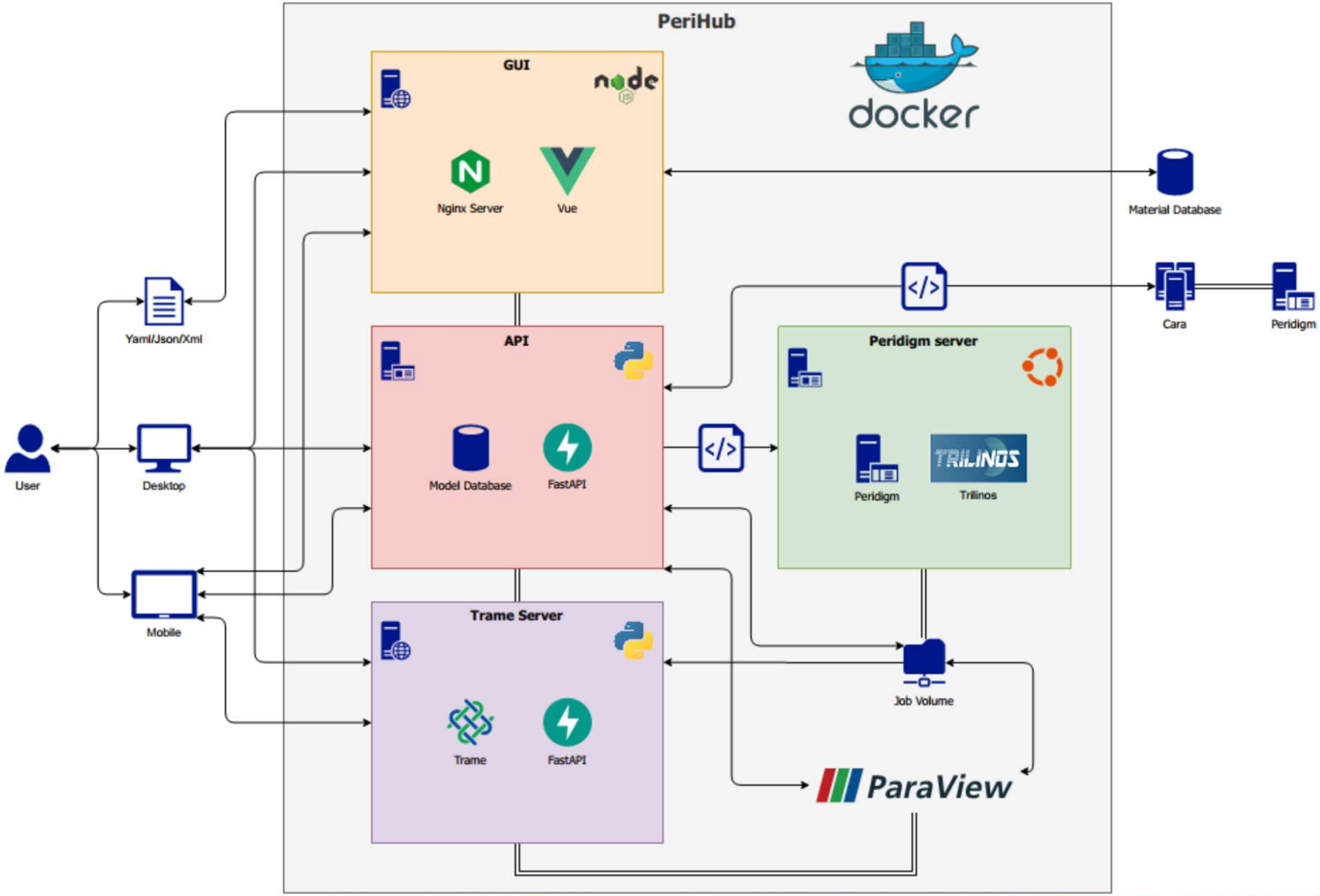
Abaqus



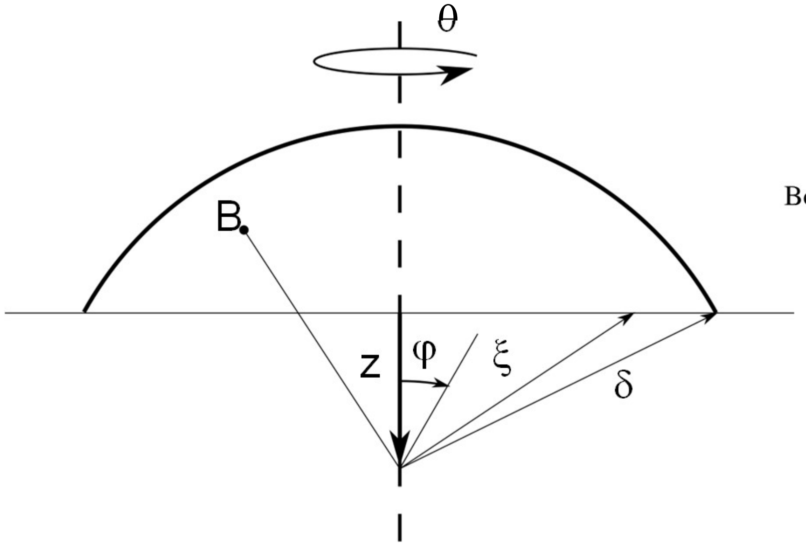
Peridigm

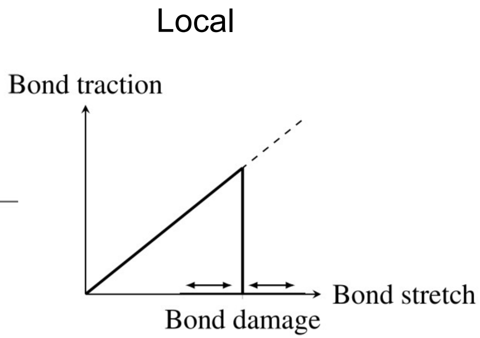


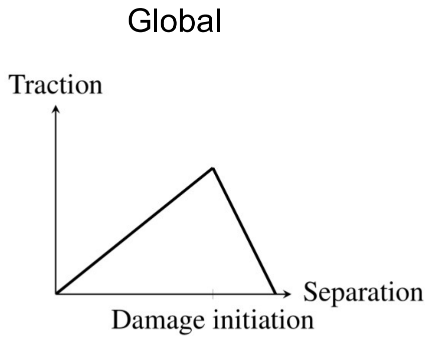
Usability


PeriHub
docker
GUI
Nginx Server
Vue
API
Model Database
FastAPI
Trame Server
Trame
FastAPI
Peridigm server
Peridigm
TRILINOS
Trilinos
Job Volume
ParaView
Material Database
Cara
Peridigm
Yaml/Json/Xml
User
Desktop
Mobile

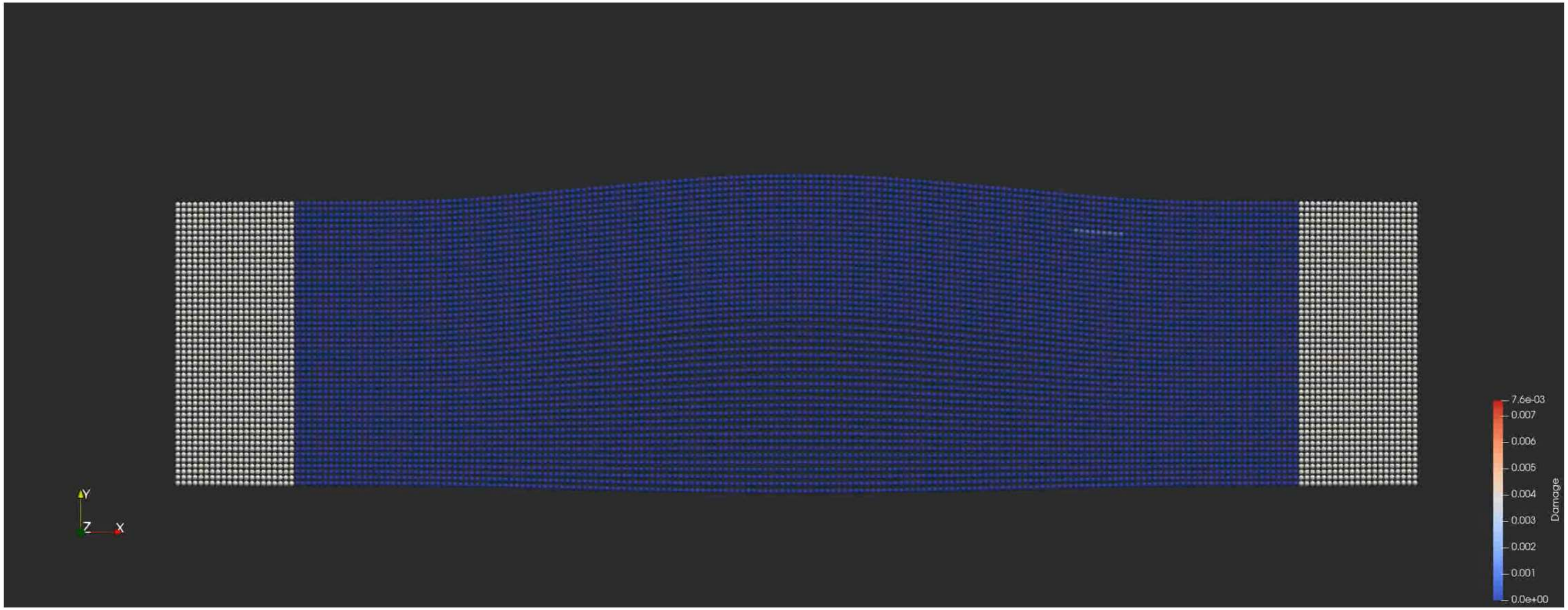
Modeling







$$G = 2\int_0^{\delta}\int_{H_r} w_c dV_\xi \qquad G = \int_0^{\delta}\int_0^{2\pi}\int_z^{\delta}\int_0^{\cos^{-1} z/\xi} w_c \xi^2 \sin\varphi \, d\varphi d\xi d\theta dz \qquad w_c = \frac{4G}{\pi\delta^4}$$



Y
Z
X
7.6e-03
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.0e+00
Damage